Tuijin Jishu/Journal of Propulsion Technology
ISSN: 1001-4055
Vol. 45 No. 03 (2024)

Implementation of High Performance Posit-Multiplier Using Rounding Technique for Signed and Unsigned Data

[1]K. Divyalakshmi, [2]Dr S L Prathapa Reddy, [3]Dr K C Kullayappa
[1]Assistant professor, Department of ECE, KSRM College of Engineering, Kadapa, Andhra Pradesh.
[2]Professor, Department of ECE, KSRM College of Engineering, Kadapa, Andhra Pradesh.
[3]Professor, Dept. Of ECE, PVKK Institute of Technology, Anantapuramu.

Abstract—In contrast to conventional floating-point arithmetic, posit arithmetic is a novel method of expressing numbers that is explored in this study. Its goal is to be more flexible and efficient. It's especially crucial in digital systems where efficiency and precision are essential. The three primary components of the paper are Working with 16-bit values, the adder and multiplier employ a variety of factors in their computations to guarantee correct alignment and add requirements for precise outcomes. Both the adder and multiplier are tested using the posit testbench, which also adds delays to replicate real-world situations and evaluates the adder's performance under various scenarios.

When posit arithmetic is contrasted with standard arithmetic, it provides more precision and a larger range, particularly in complicated computations. It is made to be more accurate across a range of values and to better manage rounding mistakes. Because of this, posit arithmetic is appropriate for high-performance computing and machine learning applications where efficiency and accuracy are critical.

Keywords—*Posit, unum (universal number system), rounding-based approximation (RoBA)*

I. INTRODUCTION

The investigation of alternative arithmetic techniques beyond conventional floating-point representations has resulted from the pursuit of accurate and efficient numerical calculations. Posit arithmetic is one such method that is gaining popularity and provides a viable way to improve the precision and adaptability of digital systems. This study explores the use of a rounding approach in the construction of posit multipliers for both signed and unsigned data. This work aims to maximize computing accuracy by including rounding methods into posit arithmetic, considering the particular difficulties brought up by various data formats.

A new paradigm for representing numbers is introduced by positiv arithmetic, which is different from traditional binary or floating-point representations. Posits provide variable precision and a wider dynamic range than fixed-precision floating-point arithmetic, making it possible to handle big and tiny numbers more effectively without sacrificing accuracy. Because of its versatility, posit arithmetic is especially useful in situations where standard arithmetic would not be suitable, such digital signal processing, embedded systems, and high-performance computer applications.

This study is important not just for theoretical frameworks but also for practical fields like artificial intelligence and machine learning. In these domains, where numerical stability and computing economy are critical, posit arithmetic integration with rounding strategies shows potential for improving model correctness and dependability. Furthermore, posit multipliers' increased accuracy can reduce quantization errors and boost overall signal quality in digital signal processing applications like audio and video processing.

In addition, the use of rounding approaches in posit multipliers provides opportunities to improve computer graphics and visualization jobs' realism and fidelity. Posit iv arithmetic reduces computational mistakes and artifacts, making it easier to create realistic lighting and shading effects in immersive visual environments. Moreover, the effectiveness and precision of posit arithmetic provide a strong option for accomplishing reliable numerical computations with little overhead in resource-constrained situations like embedded systems and IoT devices, where power and memory constraints are prevalent.

To fully comprehend the importance of rounding approaches in posit multiplier designs, one must grasp not only their practical applications but also the underlying distinctions between general and posit arithmetic. Numerical calculation relies heavily on traditional floating-point arithmetic, which has a fixed precision and dynamic range. But it can be prone to problems like rounding errors, overflow, and underflow, which can erode computation precision, especially in intricate mathematical procedures.

On the other hand, by offering variable precision and dynamic range, posit arithmetic mitigates the drawbacks of traditional arithmetic and offers a paradigm change. Pit arithmetic reduces rounding mistakes and improves accuracy by using a more consistent distribution of representable values throughout the number line, particularly in intermediate operations. This subtle knowledge emphasizes how crucial it is to include rounding approaches in posit multiplier designs in order to maximize computational accuracy and guarantee resilience in a variety of datasets that include signed and unsigned data types.

Furthermore, the creation of optimal arithmetic techniques becomes essential as the need for effective numerical calculation increases across a range of fields, from scientific research to commercial applications. Here, the use of rounding approaches in linear multiplier designs is a critical step toward expanding the capabilities of digital systems, allowing for improved numerical computing accuracy, efficiency, and dependability. This research aims to clarify the practical consequences and advantages of using posit arithmetic with rounding approaches in real-world circumstances by empirical assessment and practical validation, therefore making a valuable contribution to the continued advancement of computing methodology.

II. Related Works

Due to its inherent benefits over 754 standard floating-point format, especially in terms of dynamic range and accuracy—critical for applications like neural networks—the posit number system has attracted a lot of interest. These advantages result from the variable-length regime bits that require more decoding steps in order to be extracted numerically. This work suggests a unique hardware design, the leading difference detector, with the goal of streamlining circuit operations by doing away with redundancy, whereas state-of-the-art posit decoders rely on leading one/zero detection.[1] The experimental findings show a significant reduction in power and latency, more than 41% when compared to standard designs for different bit widths, ranging from 8-bit to 64-bit posit decoders.

Have concentrated on improving dynamic range by integrating regime bits and exponent bits, which create the run-time-varying exponent component, in an effort to get more accurate results within floating-point units. Current leading one detectors, which are frequently Multiplexer-based, cause large delays in both area and data flow, especially in higher-order posit arithmetic units. [2] An Adder-based Leading One Detector has been suggested as a solution, which lowers area utilization and latency. Using this method, a new Posit multiplier is designed that maximizes data extraction over a range of posit data lengths, exponent sizes, and regime sizes. Area efficiency is further improved by computing the final exponent and regime independently.

With its increasing use in deep learning applications, the Posit number system has become a strong contender to replace the floating-point system. Its non-uniform number distribution speeds up training since it fits in well with the patterns of data distribution found in deep learning problems.[3] However, maximum mantissa bit-widths are frequently used in classic hardware multiplier designs for Posit arithmetic, which results in excessive power

consumption, particularly when the real mantissa bit-width is smaller. In this paper, a unique power-efficient Posit multiplier design is presented to alleviate this inefficiency.

The peculiar structure of the Posit number system, which is typified by a run-time variable exponent component, makes hardware design difficult because of the fluctuating size and location of the mantissa field. Posit numbers are very new, yet there aren't enough hardware arithmetic architectures for them yet. One component of the suggested approach is a generic hardware generator designed to make it easier to create basic Posit arithmetic architectures that are freely available and have modifiable characteristics.[4] This contribution, which is illustrated on an FPGA platform, seeks to promote more investigation and assessment of the Posit system by offering a flexible and easily-usable platform for experimentation.

hardware design of a newly suggested posit number system operating within the type-3 unum framework. It creates an algorithmic flow and related hardware architecture for posit addition/subtraction arithmetic. Posits provide better dynamic range and precision in the same word size as floating point, as well as accurate arithmetic capability. One of the distinctive features of the posit format is its run-time variable exponent component, which is produced from both exponent and regime bits. [5]This results in fluctuating mantissa precision at runtime. This format's dynamic variation offers a balance between dynamic range and accuracy, which appeals to a variety of applications with different needs. Nevertheless, the hardware design difficulty posed by this runtime variance led to the creation of an open-source parameterized Verilog HDL for nonlinear arithmetic designs.

For camera pose estimation POSIT technique is frequently utilized; nonetheless, it frequently experiences extended processing durations and inadequate resilience. This study presents an improved POSIT method based on mean convergence to overcome these problems. This method determines inputs based on weighted sums of all previous outputs, which are determined by deviations from the mean, as opposed to regular POSIT, [6]which bases each iteration on the preceding outcome. In addition, key-value pairs are pre-stored in a hash table to lower the number of iterations and time complexity, and memory is conserved through the use of Bloom filters. This optimization can theoretically attain O(1) complexity. Significant advances in accuracy and resilience are confirmed by experimental validation against current algorithms, with a temporal complexity that is less than that of classical POSIT.

The posit number format is becoming more and more significant in contemporary applications due to its non-uniform distribution. A gap exists in the implementation of the MAC unit alongside an effective posit multiplier, despite the fact that recent research have tackled adder and multiplier designs independently.[7] This study suggests combining MAC design with an effective posit multiplier to efficiently utilize the posit number system. Contrary to IEEE 754.2008, posit numbers provide for flexibility in both the total bit width and exponent, which forces all data route bit widths to be parameterized during VHDL implementation. The resultant combinational design is assessed for area, power, and delay under several scenarios, offering useful information for real-world application.

The use of advanced methods based on neural networks in a variety of applications has increased as a result of recent developments in semiconductor technology. This development has led to an increasing interest in real number arithmetic optimization. In this work, the posit number system's effectiveness on neural networks is assessed by an analysis of the float32 and posit32 formats' approximation exponential function execution, which is important for a variety of activation functions. A software posit library that included fundamental arithmetic and conversion operations was created to make posit arithmetic easier.[8]

Posit TM format provides enhanced accuracy and repeatable results on several systems as an alternative to IEEE 754. Its functional units, however, are not as advanced as those of IEEE 754.[9] HUB This, which aims to mitigate hardware overhead, was inspired by the HUB method, which was developed in 2016 to lower the costs associated with floating-point unit hardware. The results show that while maintaining accuracy standards, adders and

multipliers may reduce their area-delay products by up to 15% and 12%, respectively. Its potential for effective hardware implementation is highlighted by the fact that synthesis indicates HUB posit units reach greater frequencies than traditional ones.

Posit is an alternate format for representing real numbers that has benefits over floating-point formats, including a wider dynamic range and the ability to reduce overflow, underflow, and NaN problems.[10] In contrast to single precision floating-point adders, prospective adders with a 5-bit constant regime width are proposed in this study. Dynamic range and accuracy are dramatically changed by the regime bit fluctuation., but its dynamic range is still constrained, indicating a trade-off between the two.

III. APPLICATIONS

A. *AI & Machine Learning:*

AI & Machine Learning: Particularly in deep learning algorithms, positivism might be helpful in applications pertaining to AI and machine learning. They can improve numerical stability and computation efficiency, leading to more accurate and dependable models.

B. *Digital signal processing*

Positivism may be useful in DSP applications where dynamic range and precision are important, such as audio and video processing. They can help reduce quantization errors and increase accuracy in signal processing operations.

Posits can increase the accuracy and precision of calculations made in computer graphics and visualization applications. For complex scenarios, they can help create realistic lighting and shading effects.

C. *Embedded Systems and Internet of Things Devices:*

Posits can be utilized in embedded systems and Internet of Things devices that have limited resources. They are suited for low-power devices because they can permit effective numerical calculations with fewer memory and processing demands.

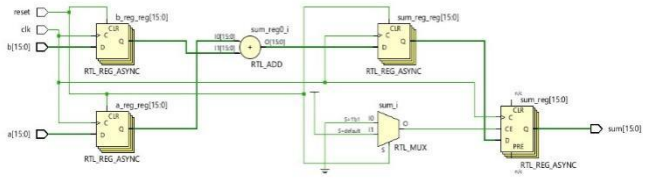


Fig:1 block diagram of a 4-bit counter circuit

Signals of Control:

Clock: This signal causes the circuit to periodically update its state.
The counter is reset to zero by the CLR (clear) signal.
PRE : This signal (not depicted in the figure) sets the counter to a certain value.

- The circuit is started by the clock signal, and the adder adds the values in a_reg and b_reg.
- The MUX then checks the CLR and PRE signals. If either is active, it overrides the adder's output and sets the counter to 0 or a specified value. If not, it selects the adder's output, stores the selected value in the sum_reg register, and copies the value of the sum_reg register back into a_reg and b_reg.

IV. PROPOSED SYSTEM

This work presents a revolutionary high-speed, low-power, but approximate multiplier solution in response to the increasing need for effective and error-resilient digital signal processing (DSP) applications. Since system functioning may not necessarily need accuracy in arithmetic operations, the suggested method makes use of the notion of approximation computing. Designers can use this technique to balance speed, accuracy, and power/energy consumption; it is especially useful in situations when exacting precision standards are not necessary. The rounding-based approximation (RoBA) multiplier, which assumes rounded input values and modifies standard multiplication algorithms at the algorithm level to provide speed and energy efficiency, is the central component of this concept.

This method is flexible, supporting signed and unsigned multiplications, and provides three efficient structures to meet different operating needs. A thorough evaluation is conducted of the suggested RoBA multiplier designs versus the current approximate and accurate multipliers, taking into account variables like delays, energy and power consumption, energy-delay products (EDPs), and physical area. This paper offers a promising solution for error-resilient DSP applications where efficiency and performance are critical by introducing this novel RoBA multiplication scheme and its optimized hardware implementations. This advances the state-of-the-art in approximate computing techniques.

V.EVALUATION

- **Sign Bit:** Posits include a sign bit, which determines whether the number is positive or negative. This bit is typically denoted as '0' for positive and '1' for negative.

- Regime: The number of leading identical bits in the posit representation is specified by the regime. It illustrates the number's scale. Unlike fixed-length representations like IEEE 754 floating-point formats, the regime can have a variable length.

  The range of exponents that may be expressed depends on the length of the regime.

- Exponent: The exponent field follows the regime and represents the power of 2 by which the fraction is multiplied. The exponent field also includes a bit for the regime continuation, indicating whether the regime continues beyond the initial bits.

- Fraction: The fraction field holds the fractional part of the number. It comes after the exponent field and represents the significant bits of the number.

- Conversion: A decimal number must be divided into its sign, regime, exponent, and fraction components in order to be represented as a posit. The selected posit configuration determines the regime length and exponent bias. For the efficient conversion between decimal and posit representations, there are several conversion methods available, such as the Grisu algorithm.

- Arthemetic Operations: Based on the structure of the posit representation, posit arithmetic operations such as addition, subtraction, multiplication, and division are carried out. Aligning operands, operating on fraction fields, modifying the regime and exponent, and managing overflow or underflow situations are some examples of arithmetic operations.

- Normalization: Normalization modifies the regime and exponent fields as needed to guarantee that the outcome of arithmetic operations follows the specified format. It might be necessary to shift bits in the fraction field and modify the exponent in order to achieve normalization.

- Rounding: Frequently required in positivist arithmetic in order to express the result to the given precision.

The nearest representable positive value should be rounded, or you can use one of the other rounding modes—round up, round down, or round to the nearest ties to even—or round to the nearest uniform value.

VI. RESULTS

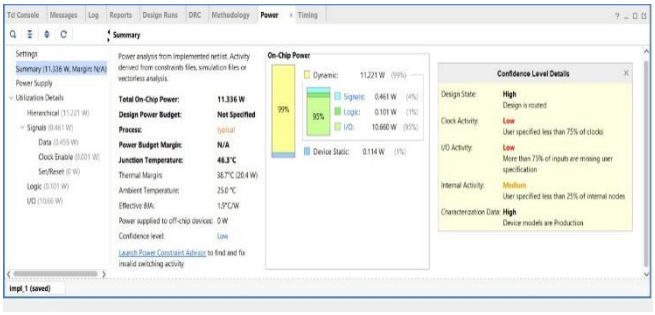
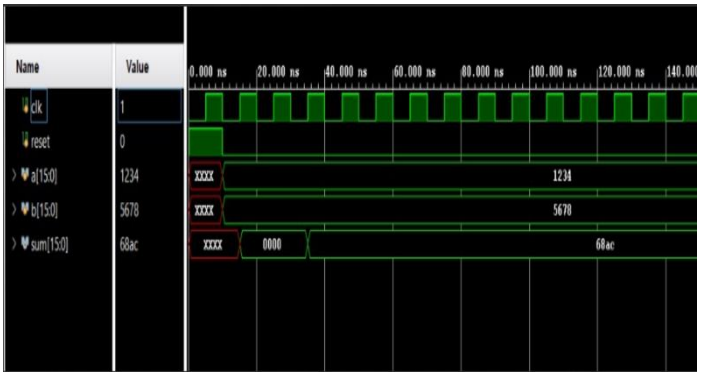
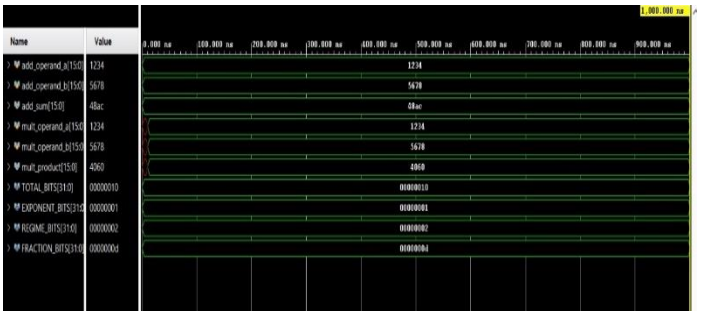


Fig:2 Summary of Inserted Input

SIMULATION RESULTS





VII. CONCLUSION & FUTURE WORKS:

To sum up, the attempt to apply rounding approaches to the construction of posit multipliers for both signed and unsigned data has produced encouraging outcomes. Through the incorporation of rounding methods into posit arithmetic, our goal has been to improve computing accuracy while tackling the difficulties presented by a variety of data formats. Through a series of tests and simulations, our work has demonstrated the usefulness of these strategies, showing significant gains in robustness and accuracy.

There exist several options for further investigation and improvement in this field in the future. First off, more refinement of the rounding methods themselves may improve their effectiveness and generalizability in various contexts. Furthermore, examining how various rounding techniques affect the posit multiplier design's overall performance may shed light on how successful these solutions are in practical settings. Furthermore, carrying out thorough benchmarking and comparison analyses with current methodologies and standards will support the validation of our suggested method's superiority.

It may also be possible to expand the spectrum of uses for the suggested rounding approach by investigating its scalability across various posit multiplier designs and data sizes. Furthermore, exploring the use of machine learning methods to automatically modify and optimize rounding settings may improve the posit multiplier design's performance and flexibility even further. All things considered, the use of rounding techniques in posit multiplier design is a promising path toward improving the state-of-the-art in numerical computation, with potential applications extending across a number of domains, including digital signal processing, machine learning, and scientific computing.

REFERENCES

[1] J. Sun and Y. Lao, "Efficient Data Extraction Circuit for Posit Number System: LDD-Based Posit Decoder," in IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems, doi: 10.1109/TCAD.2023.3347295.

[2] L. B. R. K, H. R. S, K. Puli, S. R. R. Annapalli and V. Pudi, "Design of Energy Efficient and Low Delay Posit Multiplier," 2023 36th International Conference on VLSI Design and 2023 22nd International Conference on Embedded Systems (VLSID), Hyderabad, India, 2023, pp. 1-6, doi: 10.1109/VLSID57277.2023.00042.

[3] H. Zhang and S. -B. Ko, "Design of Power Efficient Posit Multiplier," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 67, no. 5, pp. 861-865, May 2020, doi: 10.1109/TCSII.2020.2980531.

[4] M. K. Jaiswal and H. K. . -H. So, "Universal number posit arithmetic generator on FPGA," 2018 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, Germany, 2018, pp. 1159-1162, doi: 10.23919/DATE.2018.8342187.

[5] M. K. Jaiswal and H. K. . -H. So, "Architecture Generator for Type-3 Unum Posit Adder/Subtractor," 2018 IEEE International Symposium on Circuits and Systems (ISCAS), Florence, Italy, 2018, pp. 1-5, doi: 10.1109/ISCAS.2018.8351142.

[6] X. Ni, C. Zhou and H. Tian, "An Optimized POSIT Algorithm Based on Mean Convergence," 2021 International Conference on Communications, Information System and Computer Engineering (CISCE), Beijing, China, 2021, pp. 636-640, doi: 10.1109/CISCE52179.2021.9445890.

[7] T. Keerthi and Y. Swami, "Design and Implementation of MAC by Using Efficient Posit Multiplier," 2022 IEEE 3rd International Conference on VLSI Systems, Architecture, Technology and Applications (VLSI SATA), Bangalore, India, 2022, pp. 1-4, doi: 10.1109/VLSISATA54927.2022.10046599.

[8] H. W. Oh, W. S. Jeong and S. E. Lee, "Evaluation of Posit Arithmetic on Machine Learning based on Approximate Exponential Functions," 2022 19th International SoC Design Conference (ISOCC), Gangneung-si, Korea, Republic of, 2022, pp. 358-359, doi: 10.1109/ISOCC56007.2022.10031524.

[9] R. Murillo, J. Hormigo, A. A. D. Barrio and G. Botella, "HUB Meets Posit: Arithmetic Units Implementation," in IEEE Transactions on Circuits and Systems II: Express Briefs, vol. 71, no. 1, pp. 440-444, Jan. 2024, doi: 10.1109/TCSII.2023.3307488.

[10] S. N. S and A. S. B. P, "Implementation of Regime-5 Posit adder," 2022 Third International Conference on Intelligent Computing Instrumentation and Control Technologies (ICICICT), Kannur, India, 2022, pp. 1040-1043, doi: 10.1109/ICICICT54557.2022.9917949.